Array Assignments.

1. Write the code for the following problem. Assign 10 last names to an array. Write a function to display the names. Write another function to display the names in reverse order.

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Lname | Lname = [“A”,”B”,”C”,”D”,”E”,”F”,”G”,”H”,”I”,”J”] |  |
| Forward | Def forward(lname)  For I in range (0,9):  Print(lname[i]) | Last names forward |
| reverse | Def reverse(lname)  For x in range(9,0,-1):  Print(lname[x]) | Last names in reverse |
|  | Forward(lname)  Reverse(lname) | Calls the functions to print |
|  |  |  |
|  |  |  |
|  |  |  |

1. Write the code for the following problem. Add another array to problem 1 above. This array should contain exam score for the respective students. That is, the first name goes with the first score etc. These are called parallel arrays. Also modify the display functions to include exam score array in addition to the last name array.

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Lname | Lname = [“A”,”B”,”C”,”D”,”E”,”F”,”G”,”H”,”I”,”J”] |  |
| Scores | Scores = [100.0,99.0,98.0,97.0,96.0,95.0,94.0,93.0,92.0,91.0] |  |
| exams | Def exams(lname,scores):  For x in range (0,9,1):  Print(lname[x], “ scored “, scores[x]) | Prints names and their respective scores |
|  | Exams(lname,scores) |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Write the code for the following problem. The data to load is lastname and score. You can do this from a file. Add a function to problem to display the last name and highest, last name and lowest. Hint: for highest initialize a variable to 0 (high\_var). If the array value is higher than the high\_var then set high\_var to the array value and set high\_index to the position of the array. Proceed through the array until you get to the end. Do the same for finding the lowest using low\_var set to 999 (higher than the highest value).

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| File (contains data from problem #2)  Lname  score | Open file to read data  Lastname = f.readline()  Lname = []  Score = []  While lastname != “”:  Lname = string read from file  S = float from file  Score.append(s)  Lastname = f.readline()  Close file  Highandlow(lname,salary) |  |
|  |  |  |
| Highandlow  Hvar  Lvar  Hindex  Lindex | Def highandlow(lname,score)  L = len(lname)  Hvar = -1  Lvar = 999  For x in range (0,L,1):  If float(score[x]) > float(hvar):  Hindex = x  Hscore = score[x]  If float(score[x]) < float(lvar):  Lindex=x  Lscore=score[x]  Print hscore  Print lscore | Highest score  Lowest score |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Load list of 10 Player Names and Batting Averages from a file into arrays. (Create your own file with two items: player last name and batting average, i.e. 0.267, 0.300 etc). Write a function to display the arrays. Then use a while loop to repeatedly ask the user for a last name. Write another function to search for the last name in the array and then display last name and batting average when found.

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Data from file | Open file  Player = f.readline()  Playername = []  Bavg = []  While player != “”:  Playername.append(str(player).rstrip(“\n”))  B = float(f.readline())  Bavg.append(s)  Player = f.readline()  f.close()  display(playername,bavg)  response = user input string (yes or no)  while response == yes  sname = user input string sname  search(playername,sname)  I = search(playername,sname)  Print (playername[i], “Batting avg is “, bavg[i])  response = user input string (yes or no)  print (“stopping program”) |  |
|  |  |  |
| display | Def display(playername,bavg)  L = len(playername)  For x in range (0,l,1):  Print(x) |  |
| search  sname | Def search (playername,sname)  Y = len(playername)  User input sname  Sindex = -1  For y in range (0,l,1):  If playername[y] == sname:  Sindex = y  Return sindex |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Modify 4 above to display a message, “Name not found” when the name is not in the list.

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
|  | All code above is the same, the only thing modified is our while loop |  |
| While loop | response = user input string (yes or no)  while response == yes  sname = user input string sname  search(playername,sname)  I = search(playername,sname)  If I == -1:  Print (“name not found”)  Else:  Print (playername[i], “Batting avg is “, bavg[i])  response = user input string (yes or no)  print (“stopping program”) |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Example to be provided.

Load 10 employee last names and salaries into parallel arrays. Write a function to display the last names and salaries. Display the last names in reverse order. Write a function to find the employee with the highest salary. Write a loop to sum and display total of all salaries. Repeatedly ask the user for a name. Display the name and salary when found. Display message “Employee Not Found” when the last name is not in the list.